

# CNC MACHINES AND AUTOMATION

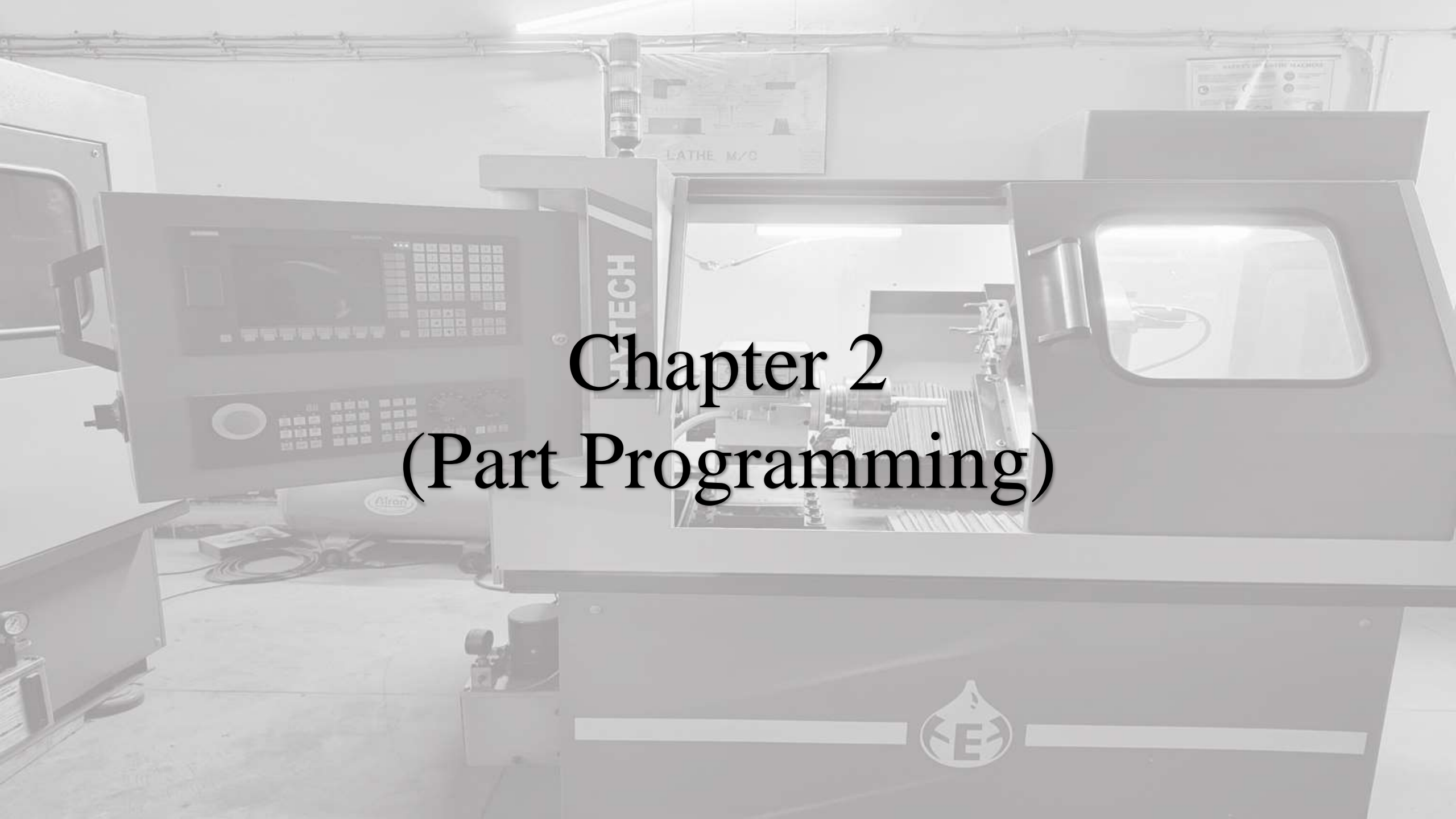


AMIT JANGRA

Lecturer

Mechanical Engineering Department

GP HISAR

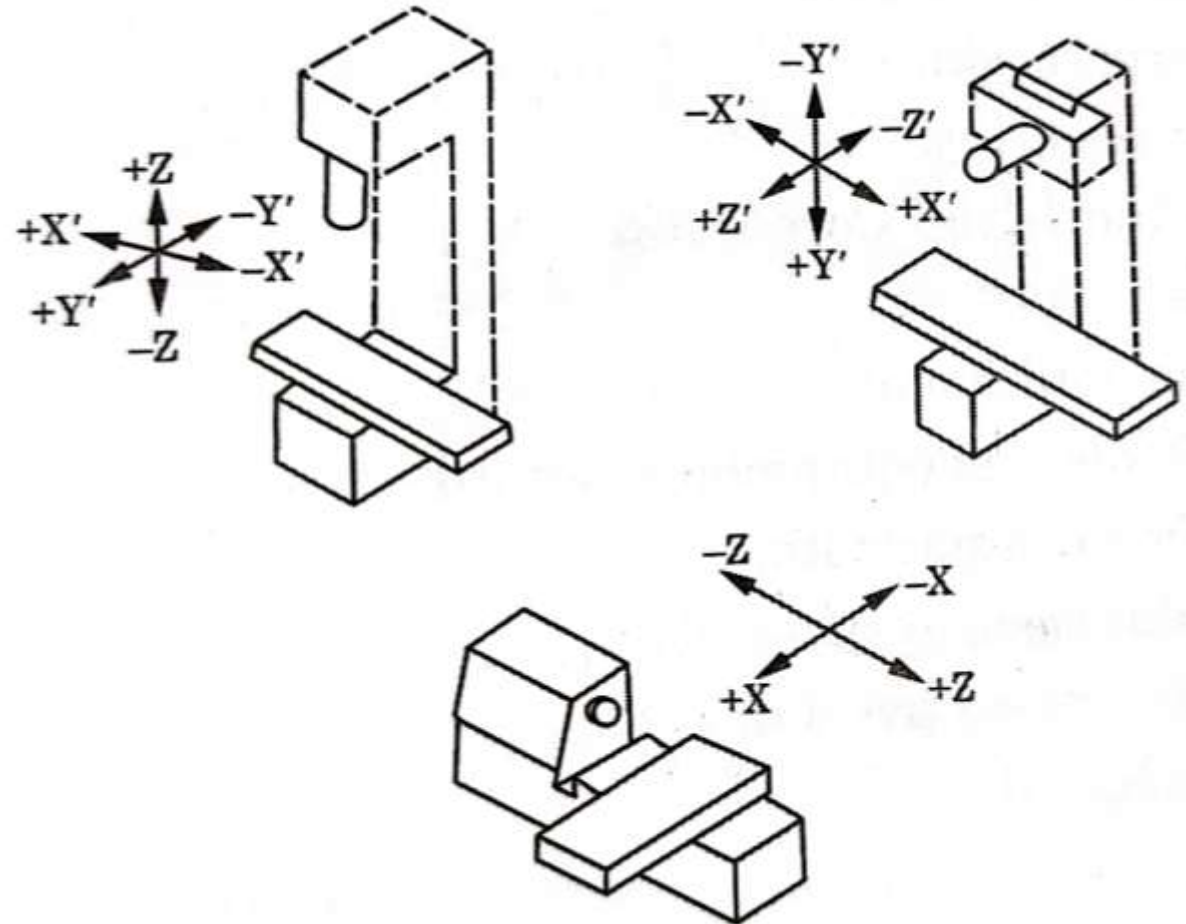
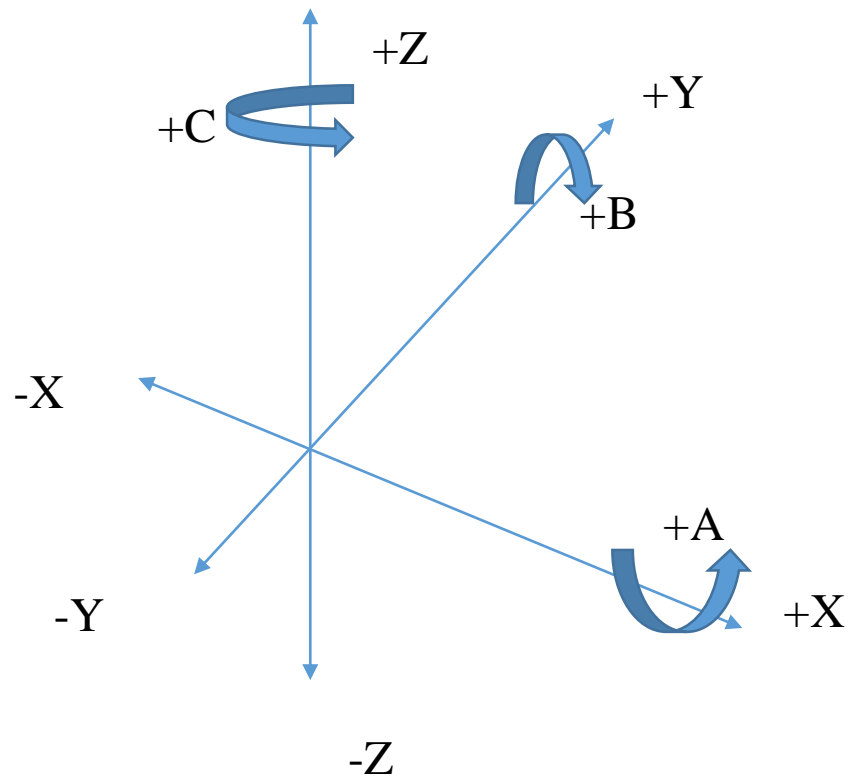


# Chapter 2

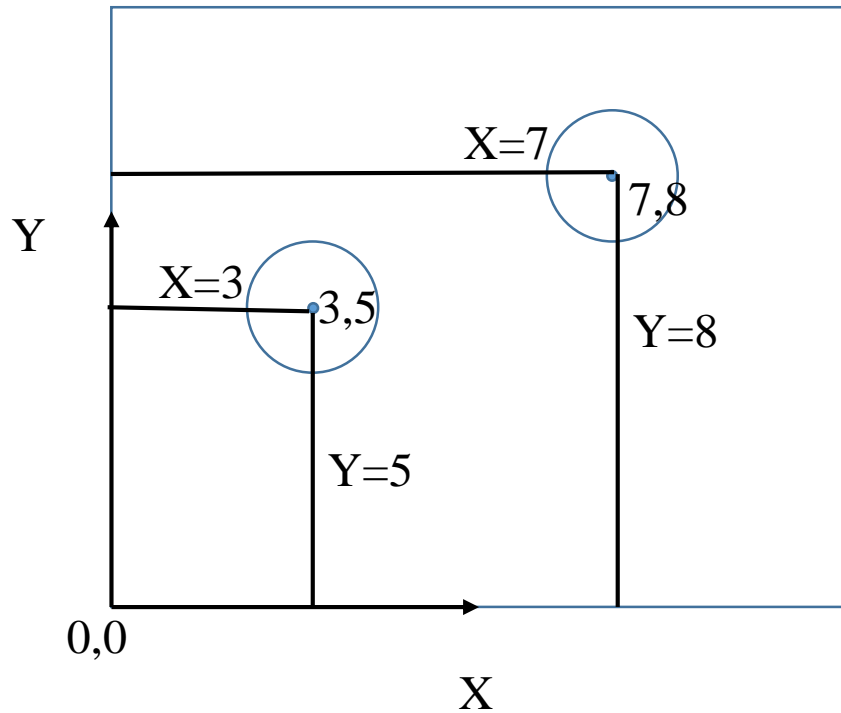
## (Part Programming)

# Axis Identification

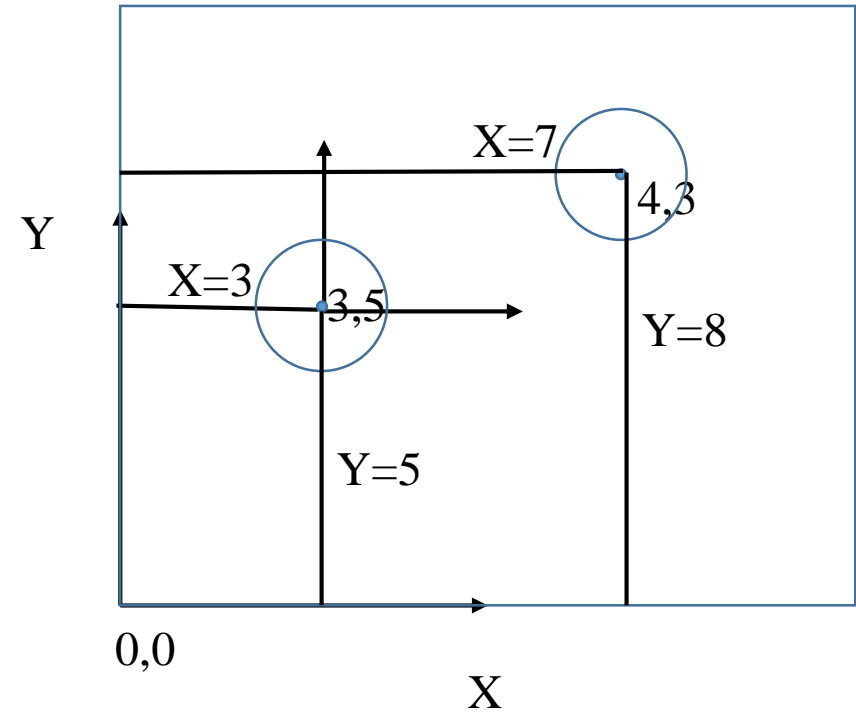
Most of machines have two or more slideways disposed at right angle to each other, along which slides are displaced.



# Main Methods of listing the Coordinate system



Absolute coordinate system



Increment coordinate system

## NC words

- i. **n-words:** - They denote the sequence number to identify the block. The complete word usually consist of three digits with 'n' as a prefix.
- ii. **g-words:** - These are called preparatory words i.e., the words used to prepare the controlling unit for the operating instructions, which are to follow.
- iii. **x, y, z, a and b words:** - They are knowns as coordinate words or dimension data words. The first three words x, y, z followed by actual dimensions, represent the coordinate position of tool along the three principal axis while the words 'a' and 'b' indicate the angular positions.

## NC words

- iv. f-words:** - These words carry the alphabet 'f' as prefix and may contain up to 8 digit maximum. They are used to specify feed rate in mm/min.
- v. s-words:** - These words carry the alphabet 's' as prefix and specify cutting speed in rev./min of the spindle.
- vi. t-words:** - These words carry the alphabet 't' as prefix and may contain up to 5 digit maximum. They are known as tool selection words and used only for those NC machines which carry a tool turret or an ATC.
- vii. m-words:** - These are known as Miscellaneous Function words. They consists of three digits as a maximum, including the alphabet 'm' as a prefix. Such function is always the last word in the block to indicate an operation.

## G Codes

Code	Description	Code	Description	Code	Description
G00*	Rapid positioning	G43	Tool length compensation in +Z	G82	Drilling cycle, counter boring
G01	Linear interpolation	G44	Tool length compensation in -Z	G83	Peck drilling cycle
G02	Circular interpolation CW	G49*	Tool length compensation cancel	G84	Tapping cycle
G03	Circular interpolation CCW	G52	Local coordinate system setting	G84.2	Rigid tapping cycle
G04	Dwell, Exact stop	G53	Positioning in machine coordinate	G85	Boring cycle
G09	Exact stop	G54*	Work coordinate system 1 select	G86	Boring cycle
G10	Programmable data input	G55	Work coordinate system 2 select	G87	Back boring cycle
G11*	Programmable data input cancel	G56	Work coordinate system 3 select	G88	Boring cycle
G17*	XY plane selection	G57	Work coordinate system 4 select	G89	Boring cycle
G18	ZX plane selection	G58	Work coordinate system 5 select	G90*	Select absolute command
G19	YZ plane selection	G59	Work coordinate system 6 select	G91	Select incremental command
G20	Select inch unit	G61	Exact stop mode	G92	Programming of absolute zero point
G21	Select metric unit in mm	G64*	Cutting mode	G93	Inverse time feed
G27	Reference point return check	G65	Macro call	G94*	Per minute feed
G28	Return to reference point	G66	Macro modal call	G95	Per revolution feed
G29	Return from reference point	G67	Macro modal call cancel	G96	Constant surface speed control
G30	Return to 2 nd reference point	G73	High speed peck drilling cycle	G97*	Constant surface speed control cancel
G33	Thread Cutting	G74	Counter tapping cycle	G98*	Return to initial point in canned cycle
G40*	Cutter compensation cancel	G76	Fine boring cycle	G99	Return to R point in canned cycle
G41	Cutter compensation left	G80*	Canned cycle cancel		
G42	Cutter compensation right	G81	Drilling cycle, spot boring		

# M-Codes

**M00** Program Stop

**M01** Program Optional Stop

**M02** End the Program

**M03** Spindle On Clockwise, Laser, Flame, Power ON

**M04** Spindle On Counter Clockwise

**M05** Spindle Stop, Laser, Flame, Power OFF

**M06** Tool Change

**M08** Coolant On

**M09** Coolant Off

**M10** Reserved for tool height offset

**M13** Spindle On, Coolant On

**M30** End the Program when macros are used

**M91** Readout Display Incremental

**M92** Readout Display Absolute

**M97** Go to or jump to line number

**M98** Jump to macro or subroutine

**M99** Return from macro or subroutine



## Part Programming

The **part program** is a sequence of instructions, which describe the work, which has to be done on a part in the form required by a computer under the control of **numerical control** computer program.

### **Basic Concept of Part Programming**

The coded instruction or commands in the form of numbers, alphabets and symbols listed in logical sequence and are fed to the controller unit of machine tool to perform a series of operations. This set of coded instruction is called part program.

# Part Programming

## **Basic Concept of Part Programming**

Following steps are needed to perform while performing a part program:

- i) Determine the start up procedure, which includes the extraction of dimensional data from part drawings.
- ii) Select the tool and determine the tool offset.
- iii) Set up the zero position for the work piece.
- iv) Select the speed and rotation of the spindle.
- v) Set up the tool motions according to the profile required.
- vi) Return the cutting tool to the reference point after completion of work.
- vii) End the program by stopping the spindle and coolant.

# Fundamental of Part Programming

- i) Process planning
- ii) Axes selection
- iii) Tool selection
- iv) Cutting process parameter
- v) Job and tool setup planning
- vi) Machining path selection
- vii) Part Program writing

# **Basic Terms of Part Programming**

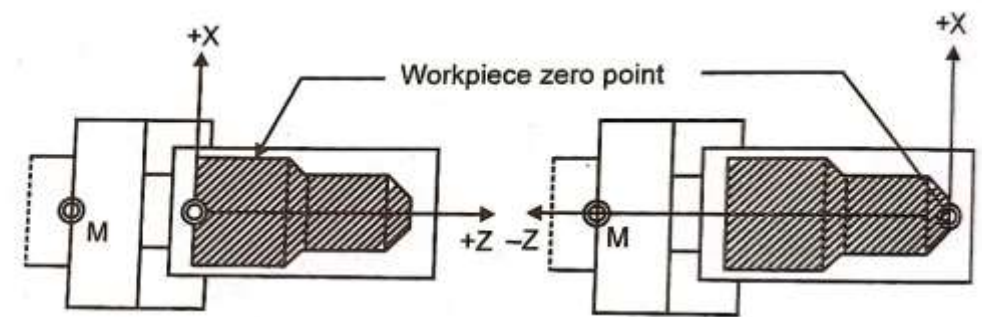
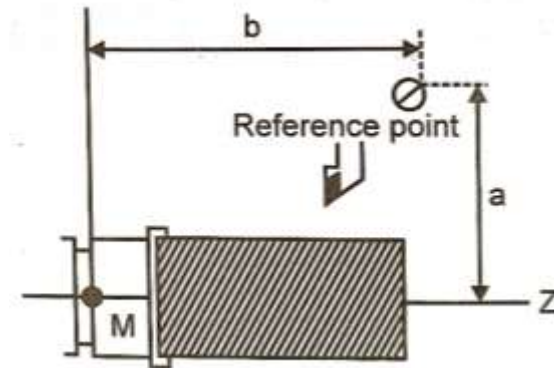
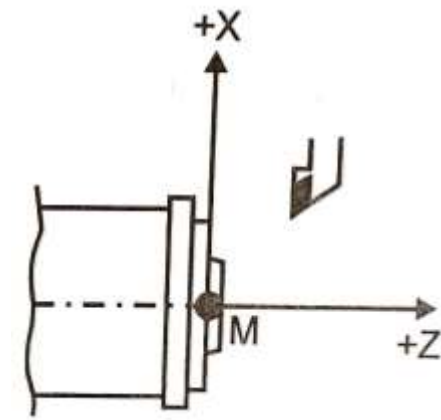
- i) Part Program
- ii) Main Program and structure
- iii) Input unit
- iv) Coordinate system

# Zero Point

Machine tool zero point

Reference point

Workpiece zero point



# Machine tool Zero Point Setting

- i) Manual Setting:** - The operator can use MCU controls to locate the spindle over the desired part zero and then set X and Y coordinate registers on the console to zero.
- ii) Absolute zero shift:** - This method can change the position of the coordinate system by a command line in the CNC program.  
**N1 G28 X0 Y0 Z0** (sends spindle to home zero position)

# Part Programming Format

## Tab sequential format (NC only)

N010 G00 X100.00 Y200 Z10 F30 M08  
010 > 00 > 100.00 > 200 > 10 > 30 > 08

## Fixed Format (NC only)

N010G01 X10 Y20 Z30 F30 S1000  
010 01 10 20 30 30 1000

## Word address format (NC & CNC)

This format is standardized by EIA and there are no TAB codes used.

N01 G01 X30 Y20 Z10 S500 F80  
T01 M01

## Compatible format (NC & CNC)

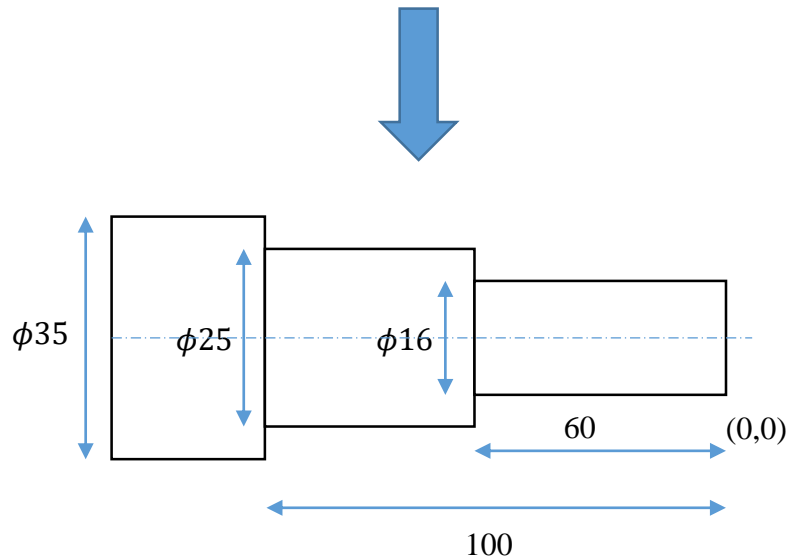
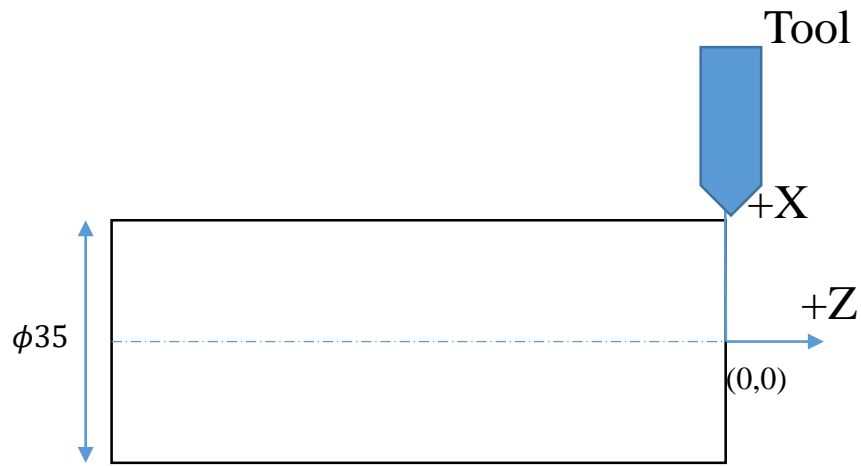
It is similar to word address format, but TAB codes are added in it.

**Part Programme Structure**

<u>N03</u>	<u>G02</u>	<u>X300Y200Z10</u>	<u>I100</u>	<u>J-10</u>	<u>K20</u>	<u>S450</u>	<u>F80</u>	<u>T03</u>	<u>M01#</u>		
Block No	Preparatory Code	Location along X-axis	Location along Y-axis	Location along Z-axis	Center position along X-axis Curved paths	Center position along Y-axis Curved paths	Center position along Z-axis Curved paths	Spindle speed	Feed speed	Tool Specification	Miscellaneous code



# Simple Programming for Rational Components



```
% 1000;  
N01 G54 G90 G71 G94;  
N02 G74 X0 Z0;  
N03 M03 S800;  
N04 G00X36
```

```
N05 Z2;  
N06 Z-0.5  
N07 G01 X-1 F0.5;  
N10 G00 Z1 X36.0;  
N15 X30;  
N20 G01 Z-100;  
N25 G00 X36 Z1;  
N30 X25;  
N35 G01 Z-100;  
N40 G00 X36 Z1;  
N45 X20;  
N50 G01 Z-60;
```

Part Programme of Turning operation

(Main Programme)

(Parameter Settings)

(Return tool at home Position)

(Spindle rotate clockwise at 800 rpm)

(Rapid travel of tool near workpiece at a safe distance first in X)

(then in Z)

(Depth of cut in z for facing)

(Facing the job)

(Retrieval of tool)

(Depth of cut in X)

(Turning in Z)

(Clearance position)

(Depth of cut in X)

(Turning in Z)

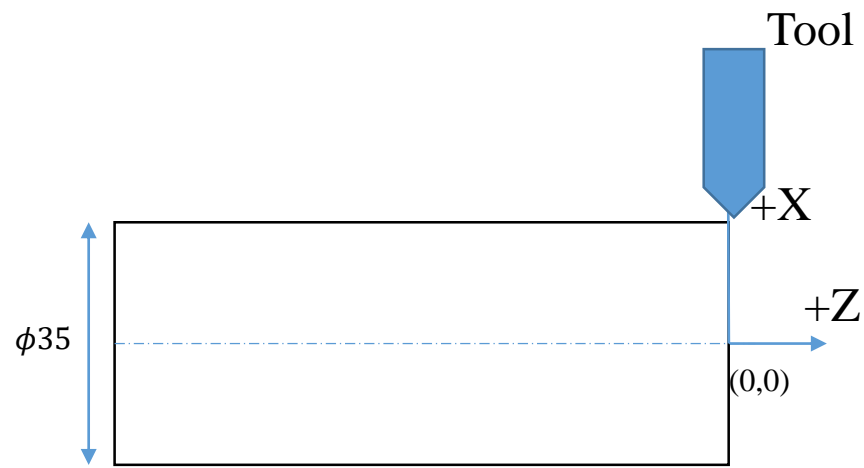
(Retrieval of tool)

(Position of cut)

(Cutting length)

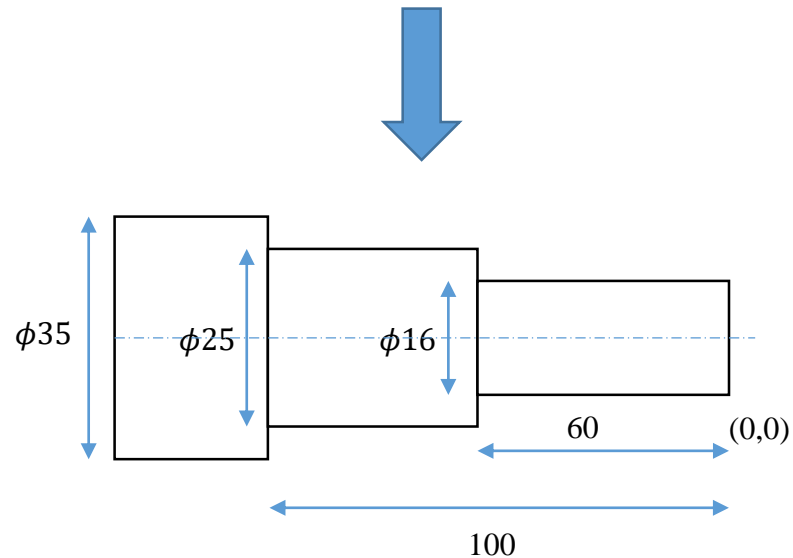
Cont...

# Simple Programming for Rational Components



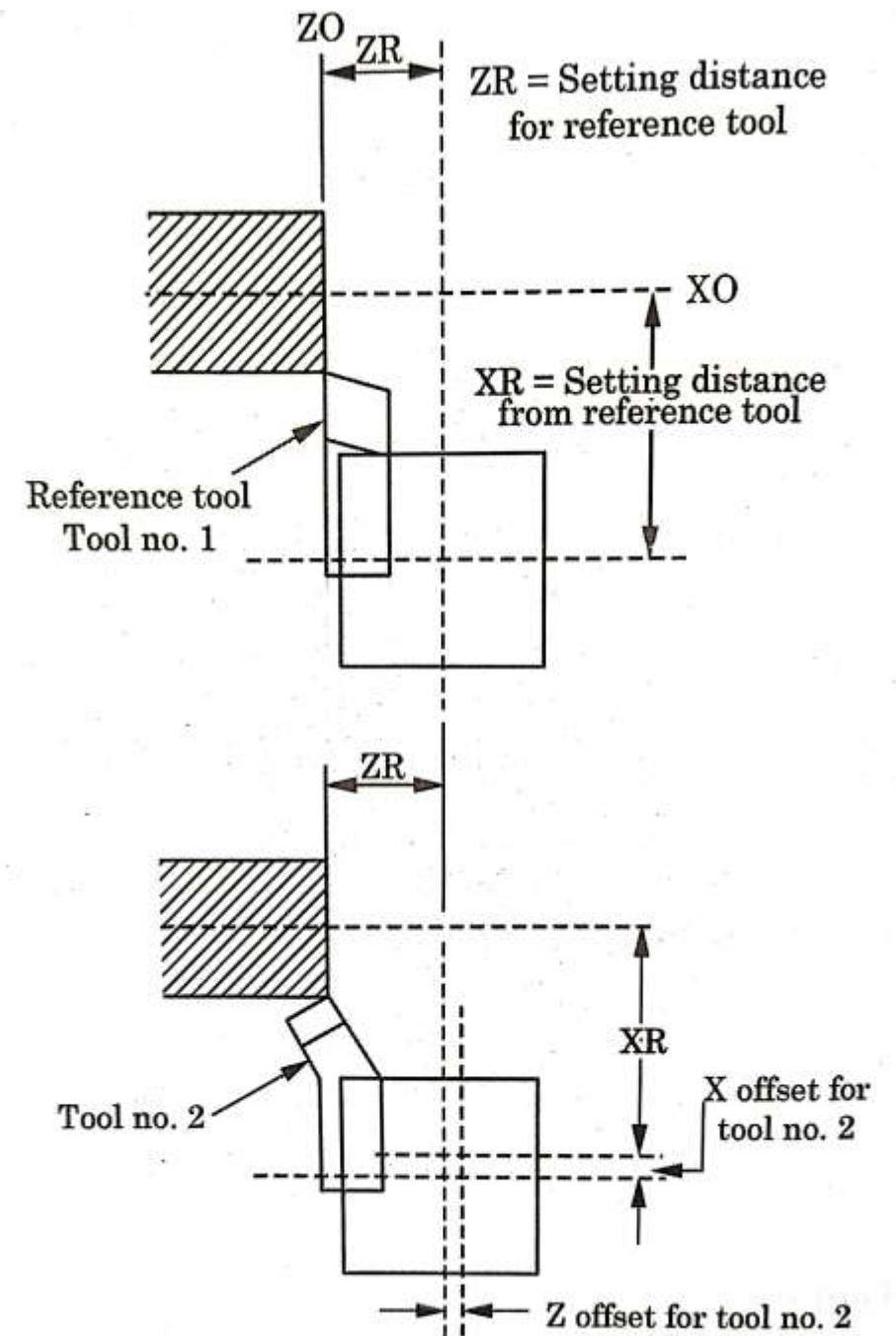
```
% 1000;  
N50 G01 Z-60;  
N55 G00 X36 Z1;  
N60 X16;  
N65 G01 Z-60;  
N70 G00 X36 Z1;  
N75 G74 X0 Z0;  
N80 M05;  
N90 M30;
```

Part Programme of Turning operation  
(Main Programme)  
(Cutting length)  
(Retrieval of tool)  
(Position of cut)  
(Cutting length)  
(Retrieval of tool)  
(Return tool at home Position)  
(Spindle Stop)  
(End of Programme)



**Tool Offsets:** Correction for dimensions of tools and movements of the workpiece has to be incorporated to give the exact machining of component. This is known as tool offset.

Normally, it is found that the size of the workpiece is not within the tolerance due to wear of the tool. It is then possible to edit the value of offsets to obtain the correct size, this is known as tool wear compensation.



# Tool Compensation

```
graph TD; A[Tool Compensation] --> B[Cutter radius Compensation]; A --> C[Tool wear compensation]; B --> D[This code command allows the programmer to ignore the cutting tool's radius or diameter during programming]; C --> E[Similar to the cutter radius compensation, tool wear compensation is also used in part programming.];
```

## Cutter radius Compensation

This code command allows the programmer to ignore the cutting tool's radius or diameter during programming

## Tool wear compensation

Similar to the cutter radius compensation, tool wear compensation is also used in part programming.

**Canned Cycles:** - Canned cycle or fixed cycle may be defined as a set of instructions, inbuilt or stored in the system memory, to perform a fixed sequence of operations. A canned cycles defines a series of machining sequence for drilling, boring, tapping etc. The canned cycle G81 to G89 are stored as subroutines L81 to L89. These cycles are used for repetitive and commonly used machining operations.

**Sub Routines:** - These are also known as subprograms, a very powerful saving method. The subroutines provide the capability of programming certain program that are repeated frequently. They are independent programmes that can be called any time and any number of times.

**Do Loops:** - The Do loops gives the facility to programmer to jump back to an earlier part of programme and execute the intervening programme and not separately like subroutines. It is given in the main program itself.

Refer to book for more details

# Type of Part Programming

```
graph TD; A[Type of Part Programming] --> B[Planning Departmental Programming]; A --> C[Workshop Programming]; A --> D[Manual Programming]; A --> E[Computer aided programming]; A --> F[High level programming languages];
```

Planning  
Departmental  
Programming

Workshop  
Programming

Manual  
Programming

Computer  
aided  
programming

High level  
programming  
languages

Thank You